



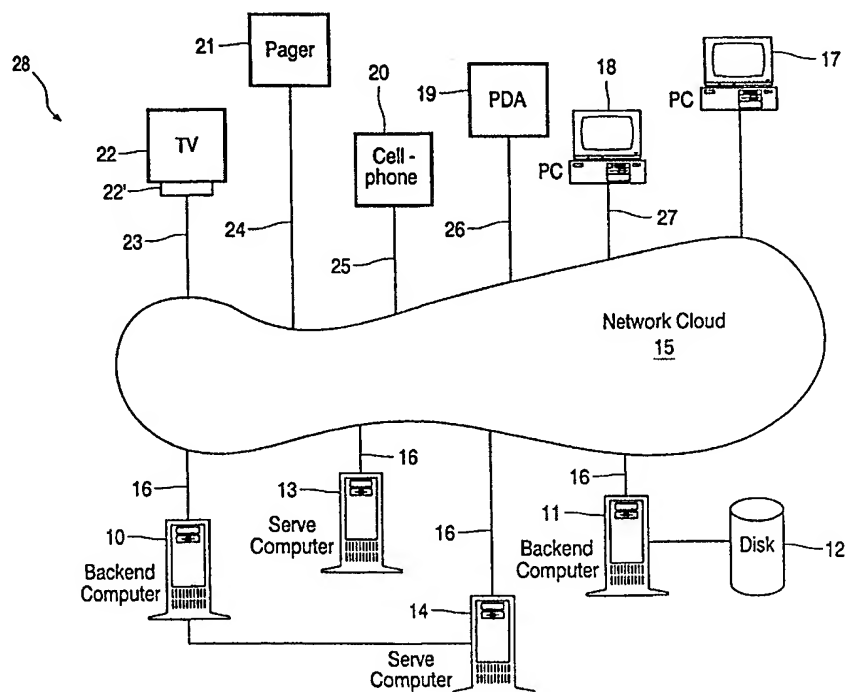
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 19/00	A2	(11) International Publication Number: WO 00/57339 (43) International Publication Date: 28 September 2000 (28.09.00)
(21) International Application Number: PCT/EP00/02753 (22) International Filing Date: 23 March 2000 (23.03.00) (30) Priority Data: 09/275,367 24 March 1999 (24.03.99) US (71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL). (72) Inventors: MOSHFEGHI, Mehran; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). PRIOR, Fred; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). WANG, Jung; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). (74) Agent: HOEKSTRA, Jelle; Internationaal Octrooibureau B.V., Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).		(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>Without international search report and to be republished upon receipt of that report.</i>

(54) Title: SYSTEM AND METHOD FOR PRESENTATION OF COMPUTERIZED PATIENT RECORDS ACROSS A NETWORK

(57) Abstract

This invention includes systems and methods for the networked distribution and personalized presentation of computerized patient record ("CPR") information. In particular, the invention includes networked computer systems structured into third-tier systems, on which reside data server-objects, second tier systems, on which reside business-server objects, and first-tier systems, on which reside object-oriented user interface applications. The business-server objects read data from the data-objects, personalize and format the data, and present it to the user interface application. Data personalization and formatting follows recommendations returned from rules modules processed by a rules engine using data from a personalization database. The rules modules advantageously centralize the rules, instead of leaving the rules distributed and embedded throughout the application logic. Additionally, decision support services can be integrated into the system by the addition of necessary rules modules to the system knowledge base.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

System and method for presentation of computerized patient records across a network.

1. Field of the Invention

This invention relates to networked computer systems, in particular to networked computer systems with object-oriented software for the personalized presentation and formatting of computerized patient records.

5

2. Description of Related Art

Medical institutions generate voluminous records of a wide variety of types in the course of patient treatment. Computer storage and distribution of these records are needed to address the well-known problems with paper records and image in integrated medical case delivery systems. Computerized patient records ("CPR") are inherently multimedia, ranging as they do from alphanumeric data, such as free text and structured reports, to non-alphanumeric data of many types, such as medical images, video streams, monitoring signals, voice dictation, and other graphics. For example, CPR information for a given patient includes demographics, admission, discharge and transfer reports, laboratory results, radiology reports, images, ordered medications, operative and procedure notes, progress notes, the status of orders and procedures, and so forth.

Access to distributed data has been hugely expanded by the great success of the internetworking suite of communication protocols, which have succeeded in linking together diverse networks of devices ranging from personal digital assistants to supercomputers. Data resident on all these devices has suddenly become remotely accessible from virtually anywhere. Further, World Wide Web protocols running on the public Internet have created unprecedented capabilities for uniform access to and standardize distribution of these vast stores of computerized data.

Further, creation and operation of distributed processing applications have been simplified by distributed object-oriented technologies. Object-oriented programming methodologies are known to have improved and streamlined the application development process. Distributed object-oriented technologies now are welding together diverse object-oriented application across networks, including TCP/IP-based networks such as the Internet,

by creating a software bus structure for uniform exchange of processing requests. Exemplary of such distributed object-oriented technologies are the standards promulgated by the Object Management Group (Newton, MA) ("OMG").

Application programming (programs written to solve specific business problems) has also been enabled by the use of rule-based techniques. An example of such techniques is the representation of an organization policies and practices by business rules. However, writing, interpreting and maintaining business rules has often been difficult for the business analysts that are responsible for specifying business rules. For example, business rules have been historically embedded directly in the procedural flow of application programs. This approach has the disadvantage that it is very difficult to change the rules to cope with changing business conditions. Business rules have also been implemented in database procedures known as database triggers. Database triggers and stored procedures are modular and isolate the business rules from the application logic. However, they require SQL programming. It is often difficult to interpret the business logic by looking at the SQL code. Database triggers are also hard to maintain and change for adapting to changing business policies.

Finally, efficient and effective integrated storage, management, distribution, and presentation of all the various forms of medical data in a flexible and evolving manner is goal that has eluded current medical information systems. What is needed are methods and systems that address the problems of computerized medical records by making effective use of the new hardware and software technologies such as those mentioned above.

Citation or discussion of a reference herein, or throughout this specification, is not an admission that such reference is prior to the invention of the subject matter subsequently claimed.

Accordingly, an object of this invention is methods and systems for efficient and effective integrated storage, management, distribution, and presentation of all the various forms of medical data in a flexible and evolving manner by coordinated and novel use of the technologies of networked computer systems, distributed object-oriented technologies, and rules-based processing for data distribution, personalization, and presentation. The methods and systems of this invention are not limited to medical institutions, but can also be advantageously employed outside of the medical arena.

Method and systems are described for personalization and formatting the presentation of CPR data depending on a user's role in a medical institution and a user's end-user devices, from personal digital assistants to powerful workstations. Personalized visualization of CPR information provides users with relevant information that is targeted for them and their end-user devices. For example, clinical users can use the present invention to access medical records quickly from a variety of networked end-user devices across the medical enterprise as well as at home and traveling.

The personalization and formatting of CPR data is performed in accordance with recommendations returned from a plurality of business rules which take into account the policies and practices of an institution as well as system and device capabilities. According to this invention, these rules and other policy and business logic are centralized in the rules engine and rules modules. This is different from the conventional approach of embedding business rules as code in application programs.

Centralizing business rules in a rules engine has benefits including allowing the medical institution to react quickly to operating and regulatory conditions. Separating business logic from application logic allows organizations to change business policies without rewriting or recompiling application code. This architecture is particularly suitable for distributed environments because it eliminates the need to upgrade client software every time there is a change in business policy. This architecture can also better support changing rules because personalization and decision support logic resides outside of the application logic. Other distributed applications using complex business logic are also well suited for the methods of this invention.

In a preferred embodiment, this invention utilizes a networked computer system functionally differentiated into a three-tiered architecture and linked by distributed object-oriented technology, such as the Common Object Request Broker ("CORBA") of the OMG. The invention includes rules-based business-server objects, a plurality of rules modules, and a rules engine.

The rules-based business-server objects receive user requests for data or results, load relevant rules and pass them to the rules engine, and process the user requests by following the recommendations returned from the rules engine. Different rule-based server objects are implemented for different rule modules to improve scalability and maintenance. The rule-based business-server objects have static or dynamic CORBA interfaces.

The rules are generally divided into personalization and decision support rule families which are centrally stored as rule modules accessed only by the business-server

objects and the rules engine server. Use of many modules improves performance and avoid undesirable interaction between certain rules. The personalization rule modules preferably include an orders and forms rule module, a printing rule module, a help rule module, and a GUI layout rule module. The decision support rule modules preferably include an alerts and reminders rule module, a drug-drug interaction rule module, and a diagnosis and interpretation rule module.

The rules engine is adapted to process the rules and return processing recommendations to the business-server objects. In a preferred embodiment, the system is coded in Java[™]. The end-user devices interface to the user by Java[™] applets inside a browser or as a standalone application.

In a first embodiment, the present invention includes an object-oriented system for computerized patient record (CPR) presentation to a user at an end-user device, the object-oriented system for implementation on computers connected by a network, the system comprising: one or more medical-records-server objects comprising CPR-request methods that input requests for CPR data, access requested data in CPR databases, and return requested CPR data, a presentation application resident in the end-user device that accepts user requests, invokes methods of business-server objects with parameters representing user requests, and displays responses returned by the business-server object methods, one or more business-server objects comprising the business-server object methods invoked by said presentation application, wherein the business-server object methods process input parameters to determine if CPR data is to be obtained, and if so which CPR data, authorize access to the determined CPR data, invoke the CPR-request methods of said medical-records-server objects to obtain authorized CPR data, format a response including any returned CPR data, and return the response to said presentation application, and wherein the authorizing and formatting are responsive to a plurality of rules retrieved from a rule database and to personalization data retrieved from a personalization database, the rules comprising (i) access-control rules for authorizing access to CPR information and (ii) presentation-control rules for guiding formatting of responses.

In a second embodiment the present invention includes a method of presenting computerized patient records (CPR) on an end-user device by an object-oriented system implemented on computers connected by a network, the method comprising: accepting a user request and invoking one or more methods of business-server objects, wherein parameters input to the business-server object methods represent the user request, and wherein said accepting and invoking are performed by a presentation application in the end-user device,

determining if CPR data is to be obtained for the user, and if so which CPR data, wherein said determining is performed by the business-server objects according to the input parameters, authorizing the determined CPR data according to authorization recommendations returned from processing both of access-control rules retrieved from a rules database and also of personalization data retrieved from a personalization database, wherein said authorizing is performed by the business-server objects, invoking one or more methods of medical-records-server objects, wherein parameters input to the medical-records-server objects represent the authorized CPR data, wherein the medical-records-server objects return the authorized CPR data retrieved from CPR databases, and wherein said invoking is performed by the business-server objects, formatting and returning a response to the presentation application, wherein the response includes returned CPR data, wherein the formatting is according to formatting recommendations returned from processing of presentation-control rules retrieved from the rules database and personalization data retrieved from the personalization database, and wherein the formatting and returning is performed by the business-server objects, and displaying the returned response to the user by the presentation application.

Other embodiments and aspects are defined in the further independent and dependent claims.

Other objects, features and advantages of the present invention will become apparent upon perusal of the following detailed description when taken in conjunction with the appended drawing, wherein:

Figs. 1A-B illustrate exemplary system infrastructure utilized by the present invention;

Fig. 2 illustrates a preferred distributed object-oriented infrastructure with rules binding;

Fig. 3 illustrates structure according of the present invention; and

Fig. 4 illustrates the general processing sequence of the present invention.

The methods and systems of this invention achieve the networked distribution and personalized presentation of computerized patient record ("CPR") data on a wide variety of first-tier, end-user devices from a variety of third-tier back-end data-storage systems by

using second-tier object-oriented server systems. The distributed, networked systems and object-oriented software infrastructures of the present invention are described first in the following. Described second are the rule-based server objects which reside on the second-tier server systems and carry out the distribution and personalized presentation on end-user devices.

Herein, CPR data is taken to include the entire gamut of medical information collected for patients at one or more medical institutions or providers and capable of computer storage. For example, CPR data for a patient can include past and present information, perhaps spanning the patient's entire life, of the following types: patient demographics; admissions, discharges and transfers; medical progress notes and discharge summaries; notes of operative and other procedures; radiology reports; laboratory results; medications; x-ray, magnetic resonance, ultrasound, and other images; the status of ordered procedures and medications; and so forth.

General Distributed Systems Infrastructure

Fig. 1A illustrates an exemplary embodiment of the distributed system infrastructure utilized by the present invention. This figure illustrates only the general classes of computers, devices, communication links, and networks utilized in the present invention; the particular connections illustrated are merely exemplary.

Therefore, as Fig. 1A illustrates, the system infrastructure includes network-connected computers functioning primarily (but not necessarily exclusively) either as third-tier back-end computers, second-tier server computers, or first-tier end-user computers or devices. Back-end computers, such as computers 10 and 11, are adapted for permanent storage of CPR data. They are advantageously provided with adequate with processing resources, main memory, and storage facilities, such as disk storage 12 which may be magnetic or optical, and with database and application software, either legacy software or software initially designed as object-oriented, for managing attached storage facilities and presenting its contents.

Server computers, such as computers 13 and 14, are adapted to host the rule-based server objects of the present invention, which are invoked by requests from end-user devices and which in turn invoke data-server objects on the back-end computers. Advantageously, server computer are provided with processing, memory, and storage resources adequate to demands of the rule-based server objects and of the distributed object-oriented infrastructure also resident on these computers.

The computers and end-user devices of this invention are networked using physical links of various types. Fig. 1A illustrates network 15 providing for general connectivity, such as the public Internet or one or more private intranets implementing the Internet suite of communication protocols including TCP/IP. Terminal links 16 to computers 10, 11, 13, and 14 can be high-speed telephone lines suitable for data server traffic. Computers can also be directly connected by local area networks ("LAN"), such as LAN 17. Finally, Fig. 1A also illustrates exemplary end-user devices 28 including: personal computers ("PC") 17 and 18, which have now standard architectures and operating systems: television set 22 with Internet interface device 22'; and highly portable end-user devices such as display pager 21, cell-phone 20 with display capability ("screen-phone"), and personal digital assistant 19 ("PDA"). Terminal network links to these end-user devices can have widely varying characteristics and bandwidth. For example, terminal PC links 27 can be switched telephone lines; terminal TV link can be a residential cable; terminal pager and cell-phone links 24 and 25 will certainly be wireless, while terminal PDA link 26 can be alternately wireless or wired.

Generally the present invention is compatible with end-user devices that both can request and display World Wide Web ("WWW") content, for example documents formatted in Hypertext Markup Language ("HTML") and distributed according to Hypertext Transfer Protocol ("HTTP"), and also have sufficient resident software to be able to interact with the distributed, object-oriented infrastructure of the present invention. As illustrated, a broad range of such devices having specialized functionality and portability are now available. Further such devices continue to be developed. After study of the subsequent description, it will be apparent to one of skill in the art that such devices can be utilized in the present invention in a routine manner simply by providing the device with a browser program capable of running Java™ applets and a Java™ object request broker ("ORB").

It will be apparent to one of skill in the art from the subsequent disclosure that, although the computers and devices preferably function in this invention primarily in one of the three roles disclosed above, because of the location transparency present in the software infrastructure, software functions and functional roles of computers can be allocated freely. For example, although not presently preferable, it is possible for all the components of this invention to reside on one computer with network attached end-user computers and devices, or even on a single computer.

Computers, including PC's, having standard architectures are suitable for use in the present invention. Fig. 1B illustrates exemplary computer 36 with one or more

processors 30, main memories 31, storage interfaces 32 with permanent storage devices 33 utilizing magnetic and optical technologies, external interfaces 34 to user displays and communication links, and buses or switches 35 interconnecting these components.

Computers and operating and database software are widely available from, *e.g.*, Sun

5 Microsystems, Compaq, IBM, Microsoft, Redhat Software, and Oracle.

End-user devices of specialized functionalities have evolving structures and operating software. Exemplary devices are available from, *e.g.*, Philips Electronics, Nokia, 3Com, and Psion, with software from, *e.g.*, Microsoft (Windows CE), the Symbian joint venture, and Sun Microsystems (Java™ Development Kit).

10 General Distributed Object-oriented Software Infrastructure

The present invention includes software objects of functions to be described, which utilize a distributed object-oriented infrastructure for communication between networked computers. Objects and object-oriented infrastructures are known to those of skill in the art. See, *e.g.*, Vogel et al., 1998, Java™ Programming with CORBA, John Wiley & Sons, Inc., New York; Object Management Group, Inc. ("OMG") (Newton, MA; and <http://www.omg.org>).

Generally, software objects are encapsulated collections of procedures, referred to as methods, and data acted on by the methods. Encapsulation means that, preferably, an object's external interface is limited only to invocations of its public methods. Accordingly, each object's data remains preferably hidden to programs other than the object's methods.

Fig. 1B illustrates exemplary objects 37 resident in memory 31 of computer 36. Objects 37 consist of collections of object data 39 and object methods 38, here denoted by op1(.), op2(.), and op3(.). Preferably, the only external interface presented by objects 37 consists of the methods op1(.), op2(.), and op3(.), and only by invoking these methods can another program gain even indirect access to object data 39.

Objects are usually designed to model real-world entities. Object data describes and models aspects of the state of the corresponding real-world entities, and object methods model actions on the corresponding real-world entities by causing changes of state and producing outputs in response to input parameters that are similar to real-world actions on the real-world entities. In other words, invoking a method produces changes object data and produces results in a manner corresponding to a real-world action performed on a real-world entity.

Accordingly, an object resident in a computer's memory causes the computer to simulate the real-world entity modeled by the object. Processing of a system structured as a plurality of client objects resident in computer memories which invoke methods of a plurality of server objects, also resident in the computer memories, cause the computers to successively simulate the client and server real-world entities modeled by the client and server objects, respectively.

Client, server, and other interacting objects need not only be co-resident on one computer, but can also be resident in the memories of a plurality of networked computers. In this case, the routing remote method invocations between client and server objects is managed by software components referred to as object request brokers ("ORB"). An ORB resident in the memory of each networked computer achieves location transparency and portability by activating necessary instances of server objects, transparently managing the communication details of remote method invocations, and optionally providing other services such as object persistence and object replication. Accordingly, the ORBS for an object-oriented software infrastructure which insulates a client object from any concern either with the identity of its computer or with the identities of computers with server objects.

The objects of this invention are preferably programmed in any programming language providing object oriented features. The C++ language is preferred; the Java™ language is even more preferred.

Various commercially available distributed object-oriented infrastructures can be used in this invention. The preferred distributed infrastructures conform to the CORBA family of standards developed by the Object Management Group, Inc. ("OMG"). CORBA-compliant infrastructures are widely available, and include products of, *inter alia*, Inprise, Inc. (<http://www.inprise.com>), IONA Technologies, Ltd. (Dublin, Ireland; <http://www.iona.com>), and ORL (<http://www.cam-orl.co.uk>). The Java™ language development kit available from Sun Microsystems also includes a Java™ ORB. Although the subsequent description of the preferred embodiment is directed to CORBA-compliant distributed object-oriented infrastructures, this invention is not so limited and can be built on other infrastructures of comparable functionality. For example, this invention can also be built with the Component Object Model and ActiveX technologies from the Microsoft Corp.

With reference to Fig. 2, certain details of the preferred CORBA-compliant, distributed, object-oriented infrastructure are described next. Included in a CORBA-compliant infrastructure is an Interface Definition Language ("IDL") implementation. IDL is a standardized, purely declarative language which program language bindings for defining

object interfaces in a object-oriented system. Compiling IDL object definitions generates client IDL stubs 103 and server IDL skeletons 105. The client IDL stubs, called by client object 109, include code to marshal a method call and its parameters into a linear form for network transmission to the server object by the communication ORBS. The server skeletons
5 include code so that server objects 110 can receive the marshaled method call and convert them into the static interfaces of the server objects.

CORBA provides two types of object interfaces. The static interface, which is more efficient but less flexible, is used if the invoked server-object methods are available as IDL definitions at compile time. The dynamic interface, which is less efficient but more
10 flexible, allows a server objects methods to be discovered at run time. Client object 109 invokes dynamic invocation interface 102 to discover how to invoke dynamically bound methods through dynamic skeleton invocation interface 106.

Object Request Broker (ORB) 100, besides being accessed through the static and dynamic interface, also has application programming interfaces ("API") to its own local
15 services. ORB core provides basic location transparency. CORBA ORBs resident on networked computers communicate using the generalized inter-orb protocol ("GIOP") over, for example, communication link 113, for routing method invocations. The Internet Inter-orb Protocol ("IIOP") is a preferred GIOP implementation using the TCP/IP communications protocol suite for communication across the Internet or private intranets.

The ORB includes component interface repository 101, implementation repository 102, and object adapter 107. The interface repository provides services for dynamically storing, accessing, and updating object-description ("metadata") information. The implementation repository is a run-time repository of information about server object classes, instantiated (in other words, activated) objects and object references, and object
25 location in the network. The object adapter records in the implementation repository object references of instantiated server objects. The ORB uses the implementation repository to activate server objects and to locate already activated objects. The object adapter provides additional services to server objects, such as support for instantiating (or, activating) server objects, passing them method invocations, and assigning them network addresses known as
30 object references.

A CORBA implementation may include standardized server objects defined in CORBA Common Services and Common Facilities standards. One such service is a standardized user authentication and security service.

Rules modules 111 and rules engine 112 are not part of the CORBA standard, and are described subsequently as part of the present invention.

System Structure of the Present Invention

Fig. 3 illustrates the preferred allocation of the software functions and objects of the present invention within the distributed object-oriented infrastructure linking the networked system computers. The preferred allocation illustrated is referred to herein as "three tier", since it includes a third tier of back-end computers, a second or middle tier of server computers, and a first tier of end-user devices or computers. Each tier is described next, followed by more detailed descriptions of the business-server objects, preferably allocated to server computers, and associated software functions and data.

As previously described, alternate object and function allocations are routinely possible because of the location transparency provided by the distributed, object-oriented infrastructure.

The First Tier - End-user Devices

The first tier includes interactive, end-user devices. Illustrated in Fig. 3 are specialized end-user device 50, which can be a Web attached TV, a personal digital assistant, or a cell-phone or pager with display capability, and so forth, and PC 51, which can be a Intel/Windows-based computer. The specialized end-devices are particularly appropriate for applications such as telemedicine or remote consultation.

In the preferred Java[™] language implementation, it is possible for software components of substantially equivalent function, principally a Java-enabled HTML browser and a Java[™] ORB, in addition to necessary system software, to reside in the memories of end-user devices, from cell-phones to PCS. Herein, an HTML browser program refers to client software that functions to retrieve and display requested HTML documents from an HTML server according to the TCP/IP-based HTTP protocol. The HTML document includes an embedded Java[™] applet, the browser automatically downloads the applet's Java[™] byte-codes from an applet store and execute them in its Java[™] virtual machine environment.

Alternatively, instead of a browser program, a separate, stand-alone program of similar function, preferably also written in Java[™], can be used together with an installed Java[™] virtual machine. Such a stand-alone program would include similar end-user device display management capabilities, screen definitions similar to any downloaded HTML pages, and internal routines with function equivalent to any downloaded Java[™] applets.

It is preferable that the browser be capable of communicating information, for example, its type and version number, so that its capabilities can be determined from stored

information, preferably in the personalization database. Similarly, the IP address of an end-user device can be used to determine its capabilities and the capabilities of its display from stored information keyed by IP address, again preferably in the personalization database.

In detail, Fig. 3 illustrates browser program 52 designed for specialized device
5 50 communicating with middle-tier server 65 over communication path 53, and browser
program 54 designed for PC 51 communicating with HTTP server 65 over communication
path 55.

Object-oriented Java[™] applets 56 and 57, running in a Java[™] virtual machine
environment provided in the respective browser programs, interact with middle-tier business-
10 server objects 67 to respond to user requests. An alternative to using the browser's virtual
machine is to use browser plug-in technology available from Sun Microsystem's
(<http://www.javasoft.com/products/plugin/>). This technology allows use of Sun
Microsystem's most recent Java[™] runtime environment instead of the browser's default
virtual machine. In particular these applets receive end-user requests, including those for
15 CPR data, determine which business-server object that can respond to the request according
to the request's type, remotely invoke methods of that business-server object, and receive the
results returned, including any CPR data. Personalized views of the requested data, for
example resident in buffers 58 and 59, are then displayed.

Data presentation and personalization tasks, which are important to this
20 invention and are subsequently described in detail, alternately either can be performed
entirely by the business-server object or can be shared between the business-object server and
the object-oriented end-user device applet in the case of more capable end-user devices.
These tasks personalize displayed results according to end-user characteristics and format
them according to the capabilities of the end-user device.

Remote invocations of middle-tier business-server object methods by the
25 object-oriented end-user device applets are automatically managed, as previously described,
by the communicating ORBs, such as Java[™] ORBs 60 and 61 resident in the end-user
memories, and Java[™] ORB resident in the memory of the middle-tier server computer 76.
The invocations can advantageously employ either the static or the dynamic CORBA
30 interface, or other CORBA facilities and features.

In the preferred embodiment, inter-ORB communication utilizes the TCP/IP-
based protocol IIOP. Since browser program communication is also TCP/IP based, all
communication between the end-user devices and the middle-tier computers, for example
over communication paths 53, 55, 62, and 63, can easily share a single physical network,

since TCP/IP protocol stacks (not shown) resident in the memory of each computer and end-user device multiplexes the communication paths onto single links.

As also described subsequently with respect to Fig. 4, the software functions illustrated in Fig. 3 in the end-user devices are established after successful end-user logon.

5 For logon purposes, the browser program retrieves HTML logon pages, obtains the user's authenticating information, and verifies it with a middle-tier security server. Preferably, the logon pages include object-oriented Java[™] applets which remotely invoke authentication methods in security-server objects 70. In one alternative, following successful authentication, one or more HTML pages are retrieved by the browser programs, direct
10 downloading of Java[™] applets 66 and 67, and commence end-user interaction.

Personal smart cards can also be used in this invention to provide certain user authentication information as well as certain personalization information (in the case where the end-user device shares personalization tasks). Smart cards are credit card sized devices which contain a processor, temporary memory, and permanent memory, a certain portion of
15 which is secured from tampering or unauthorized access. Authentication information can be stored in the secured portion, while general personalization information can be stored in the remainder of permanent memory.

The Middle Tier

Middle-tier software functions, illustrated in Fig. 3, respond to remote method
20 invocations from end-user device applets with various requested results, including partially or entirely formatted and personalized CPR data fetched from back-end data-server objects. Because of the location transparency of the distributed object-oriented software infrastructure, the various middle-tier and other object system functions can be allocated freely to one or more computers.

25 Middle-tier software components are now sequentially described. Security server 70 includes objects providing methods for user authentication. Logon HTML pages gather user authentication information, such as user-id and password. Preferably, Java[™] applets, also part of the logon pages, remotely invoke the authentication methods with the gathered authentication information as parameters. Successful authentication allows
30 completion of logon and commencement of interactive usage; unsuccessful logon blocks further response from the system. Various kinds of user authentication information can be used in this invention, including information read from a user's smart card by an end-user device card reader or biometric information obtained by a biometric scanner.

Business-server objects 67 are remotely invoked by end-user device applets 56 and 57 to process the various types of user requests, including requests for CPR data, for data entry, for alerts and reminders, and for decision support functions. Requests processing is summarized here and described in greater detail in connection with the business objects themselves. For all requests, the business-server objects proceed only after first ascertaining that the user has the authorization or access privileges needed to make the request.

First, for CPR data requests, the business-server objects, after determining user access privileges, retrieve the requested and authorized data by remotely invoking methods of data-server objects resident in the memories of back-end systems. They then, optionally in cooperation with the end-user device applets, personalize and format data for display. In one implementation, each business object demultiplexes requests for all types of CPR data, directing each to the appropriate data-server object. Here, the end-user device applet is simplified in that it needs to address a CPR data requests to one class of business-server object. In another implementation, each type of CPR data is retrieved by a separate class of business objects. Here, the end-user device applet itself must direct the request according to the type of CPR data requested.

For data entry requests, perhaps concerning a patient of an end-user, the business-server, after authorization checking, retrieves the appropriate data-entry forms or instructions from the responsible data-entry-server object, provides these to the end-user applet, and returns entered data to the responsible server object. In some implementations, the business-server object itself is responsible for data entry.

Business objects can also provide decision support capabilities to authorized users. For such requests, they can interactively obtain necessary input information from an end-user and then invoke decision support processing by, for example, medically-directed expert systems.

Business-server objects can also generate alerts and reminders for a requesting end-user. Here, in one embodiment, the business-server object reviews CPR data for patients associated with that end-user and issues appropriate alerts and reminders. Alerts can be generated by applying decision support processing to CPR data. For example, a physician can be alerted to possible drug interactions based on drug interaction decision rules applied to CPR data including current medications for a patient being treated. Reminders can simply be end-user entries placed in the patient records for later, timed display.

Finally, the system of this invention is open to the addition of other types of end-user requests in a routine manner. To do so, end-user applets can be modified to provide

for the display of the new type of request and for entry of its associated parameters, and a new business-server object can be added to the middle-tier to process the new request, perhaps in cooperation with data retrieved from back-end data-server objects.

HTTP server 65, HTML store 66, and applet store 67 have been previously described in connection with first-tier components. Rules engine 68, rule modules 71 and 72, and personalization database 69 are subsequently described in connection with the business-server objects themselves.

The Third Tier

Persistent CPR data of all types, for example those types previously enumerated and other types that may become useful, are stored on back-end server computer systems, which make the stored data available as data-server objects, such as system 77. The business-server objects of the middle-tier remotely invoke methods of data-server objects 73 resident in memories of back-end computers during the processing of their own methods, which were, in turn, remotely invoked by end-user applets resident in the memories of the end-user devices. Where the data-server objects are remote, this invocation is managed by middle-tier resident ORB 64 communicating with third-tier resident ORB 74.

Third-tier data server-objects can either be components of a system initially designed to be object-oriented or they can be legacy systems with interface software (a "wrapper") for transforming legacy programming interfaces into object-oriented method interfaces. Such object "wrapping" is known in the art. See, for example, U.S. patent application 09/096694, filed June 12, 1998. Legacy systems 75 include, for example, Picture Archiving and Communication System (PACS), Hospital Information System (HIS), Radiology Information System (RIS), Cardiology Information System (CIS), laboratory system, etc. Details of such object-wrapped, legacy systems are not shown in Fig. 1.

Business-server Objects - Rule Processing

In the next two sections, business-server object processing is presented in detail. In the following two sections, the data controlling this processing, namely the personalization database and the rules, are presented in detail.

Business-server objects processing is preferably rule based. User requests and their associated parameters are processing according to request type by following recommendations resulting from application of stored rules to end-user personalization data. The personalization data includes at least information concerning user characteristics, such as user role and user privileges, and concerning user environment, such as the type of end-user

device, the network link to the device, the device's display capabilities, browser program capabilities, and the room location.

Business-server objects use rules to guide the often complex and frequently changing criteria by which users are authorized to access system services, and to access particular CPR data items. Rules are advantageous for such decision-making because, at least, they are a known, modular, and extendable programming method for decision making. First, it is known how to express such criteria in rules, for example expert-system rules. Second, rules can be modularized so that changing decision criteria can be met by routine modification of specific modules without any need to modify code. Finally third, with appropriate additional rule modules, the same rules processing can provide decision support services.

Although rule-based processing is preferred in this invention, it will be apparent to one of skill in the art that this invention can also employ similar known, modular, and extendable decision-making processing methods, or combination of such methods, known in the artificial intelligence arts. For example, the effect of rules processing herein can be achieved by methods based on a first-order predicate logic knowledge representation and associated inference engine. See, *e.g.*, Russell et al., 1995, Artificial Intelligence - A Modern Approach, Prentice-Hall, Inc., ch. 7-10.

In the preferred embodiment employing rule-based processing, rules used by the business-server objects are preferably expressed with a limited set of programming language constructs expressed in standard syntax, including variables, boolean expression, and statement including "if-then-else" statements, assignment statements, and rule invocation statements. For example, the informal access rule that "if the user is a physician and if the user is the physician of the current patient then the user can see the records of the patient" can be expressed in a precise syntax in the following manner:

```
RULE SeeRecord {  
  Valueproperty case;  
    IF (user.function == "physician") AND  
      (patient.physician == user)  
    THEN { approved.value = true; }  
    ELSE { approved.value =false; }  
}
```

Exact rule syntax varies depending on the exact rule language and the rules engine used in a particular embodiment.

Rules semantics is simply related to the understanding of their syntax common to those of skill in the art. Each rule is typically an "if-then-else" statement, which can be optionally nested. Rule evaluation begins with evaluation of the Boolean "if-condition" based on data values input or assigned to rule variables. The "if-then-else" of the rule body is evaluated and any statements in the "taken" branch are performed. For example, performing assignment statement routinely sets the values of one or more rule variables to the value of an expression. Performing rule invocation statements routinely chain rules, so that a first rule can invoke a second rule. The rule to be evaluated next can be selected sequentially from a list of rules, or according to a priority assignment which can be dynamically updated, or by a rule invocation statement.

A rules engine evaluates rules presented to it. In one implementation, the rules engine simply interprets rules presented in a plain-text form.

Generally, rules are stored separately from the business-server objects in rule databases, or rulebases. Needed rules are retrieved from the rulebases by business-server objects or the rules engine as needed.

Further, related rules are advantageously grouped together into rule-sets; related rule-sets are also advantageously grouped together into files known as rule modules. Grouping rules into rules modules is advantageous for the following reasons. First, undesirable interactions between certain rules can be avoided by putting those interacting rules into separate modules separately presented to the rules engine. Second, rule modules and rule sets can be assigned priorities which helps to order their evaluation. Third, smaller rule modules improve performance because only a subset of the rules and object are used at any given time and need to be stored in memory. Finally, rule modules also facilitate rule update and maintenance, since each rule module can be maintained by the most knowledgeable person. For, example, administrators can update policy and practice rules; domain experts can update decision support rules; and users can update their personal rules.

Thus, instead of having one bulky cumbersome rule module for a whole medical institution, several separate rule modules representing different aspects of the institution's policies and practices are preferable. Accordingly, related rules for personalization and decision support are grouped into a plurality of smaller rule modules having different functions.

Rules used in this invention can be programmed with various commercially available knowledge-based application generators, including, *inter alia*, ART*Enterprise (Brightware), LiveModel (Intellicorp), Elements/Advisor (Neuron Data) and AionDS

(Platinum Technology). Each of these have their own specific syntax for specifying rules. Also, the Arden syntax has been developed for general uses in medical knowledge applications, and has been adopted as a standard for health knowledge representation. See, *e.g.*, <http://www.cpmc.columbia.edu/resources/arden>.

5 A preferable commercial knowledge-based system provides graphic rules editors for defining new rules and modifying existing rules, so that even non-programmers can view and update rules. Such a product also provides tools for checking the consistency of the rules.

Business-server Objects - Processing

10 General system processing, and its distribution among the system tiers, is now explained in detail with reference to Figs. 3 and 4, in which the labels "1T", "2T", or "3T" denotes processing that occurs at the first tier, the second tier, or the third tier, respectively.

 At access step 80, initial accesses of an embodiment of this invention by a user causes the browser program, which is resident in the first-tier end-user device, to contact
15 HTTP server 65 at the second tier, whereupon the primary, or root, HTML page is automatically downloaded from HTML store 66 at first download step 81. Next, at second download step 82, the end-user device applet referenced in this primary HTML page is downloaded from applet store 67, and applet execution is initialized and started in the browser program. From this step onward until end-user logoff, the object-oriented Java™
20 applet manages the user-interface in the end-user device and responds to user requests by making remote invocations of methods in second-tier server-objects.

 The first action of the end-user device applet is authentication step 83 during which the user is authenticated to the system. The end-user applet remotely invokes authentication methods in second-tier security-server objects 70. If the security-server
25 objects do not authenticate the user, appropriate security actions are undertaken to safeguard system integrity and confidentiality. Further, system security is provided by well known protocols which use digital signatures, authentication, and document alteration prevention techniques.

 If the user is authenticated, processing continues with subsequent steps 84 to
30 94, which implement the user-request-processing loop. This loop is executed until user logoff is detected at test 94. After logoff, at step 95, the end-user device returns to a neutral state.

 The user-request-processing loop begins at wait step 84, where it waits for a user request. Upon entry of a user request, after which the end-user device applet gathers the

request type and associated parameters, and determines the appropriate business-server object to service request of the input type. During remote invocation step 85, the request and parameters are passed from the first-tier end-user device applet to appropriate second-tier business-server object 67.

5 Each second-tier business-server object generally processes remote method invocations according to steps 86-90. First, at read step 86, business-server object 67 reads stored rules modules that recommend the processing of the entered request type. Rules modules of this invention are generally considered either as belonging part to the decision support rules module family 71 or to the personalization rules module family 72, each of
10 these families having the exemplary members illustrated in Fig. 3. These rules modules and related request types are subsequently described in more detail. Rules modules are read in the order of their assigned priority for efficiency and for control of the processing order. As the business-server object reads rules of each rules module, it also reads from the personalization database values for personalization data types referenced in each rule and
15 pertaining to the particular end-user. This read access is represented in Fig. 3 by paths from business-server objects 67 to personalization database 69 and to rules modules families 71 and 72.

 Therefore, at call step 87, the business-server object read rule modules and referenced personalization database values. In this step, the business-server object can read
20 both rules primarily related to authorizing and satisfying the entered user request as well as personalization and formatting rules and referenced data. Alternatively, it can defer reading the latter rules and data until they are needed at format step 90. The business-server object then calls, or otherwise arranges to pass, the retrieved rules and the user's personalization data-type values to rules engine 68 for processing. Because of rule chaining, rules initially
25 passed to rules engine 68 may requires further access to additional rules modules and personalization data-type values. This access is illustrated in Fig. 3 by paths from rules engine 68 to personalization database 69 and to rules module families 71 and 72.

 Aspects of rule-based server object processing is also illustrated in Fig. 2. Therein, methods of server object 110 illustrated read rules modules 111 and pass them to
30 rules engine 112, which returns processing recommendations to these methods. Because of rule chaining, it may be necessary for rules engine 112 to directly read rules modules 111, as illustrated. One of skill in the art will recognized that this invention is not limited to this particular rules access structure, but comprehends other access structures that make needed rule modules and personalization data type values available to the rules engine.

Returning to Figs. 3 and 4, test step 88 distinguishes two processing cases depending on the type of entered request and recommendations returned by the rules engine. In one case, no access to actual CPR data is needed. For example, in the case of a CPR data request, the rules engine may recommend denial of the user access request. In the case of a decision support request, all needed data may be entered as request parameters by the end-user.

If CPR data access is needed, at remote invocation step 89, business-server objects 67 make remote method invocations to methods in appropriate third-tier, data-server objects 73. Data-server objects 73 access CPR data, either directly from object-oriented CPR database systems or indirectly through wrapper code from legacy CPR database systems 75, which are typically non-object-oriented.

In either case, at format step 90, the data or results are personalized and formatted according to recommendations returned by rules engine 68 upon processing personalization and format related rules modules and data. These modules and data can be either read in prior step 86 or deferred to this step. For example, a formatting recommendation may be to format a medical image at a resolution of 640x480.

At return step 91, the personalized data or results are returned to the first-tier, end-user device applet, thereby completing the remote method invocation of the business-server object initiated at step 85. Finally, returned data or results are displayed to the end-user at display step 93.

For capable end-user devices, such as PC 51, certain data or results formatting operations, principally machine specific formatting operations, can be advantageously off-loaded to the end-user PC. In this case, also resident on PC 51 would be certain local rules modules and personalization data, typically machine specific formatting rules and machine characteristic data, and a local rules engine. Preferably, the local rules are coded in Java[™] and are downloaded as an applet. End-user device applet 54, before final display, would interact with the local rules engine, in a manner similar to the interaction of business-server objects 67 with rules engine 68, and would further format returned data or results according to recommendations derived from the resident rules and returned by the local rules engine. For example, an entire high-resolution medical image can be downloaded to PC 51, with final formatting at the resolution of the attached display, either as a whole or in segments as the user pans across the image, deferred to the PC. For a high-end end-user workstation, this alternative advantageously shifts processing load away from the second-tier server to first-tier end-user devices.

Personalization Database

The personalization database contains values for data of varied types that are referenced by the stored system rules that guide business-server object processing.

Exemplary data types typically present in the personalization database are discussed

5 individually as in the classes that follow.

It will be understood by one of skill in the art that the following list of classes and their categories and subcategories are not limiting, and other classes and categories of data may be advantageously stored in the personalization database.

USER ATTRIBUTES CLASS:

10 Data of the user attributes class indicates the relationship of the user to the medical institution or to particular patients, as well as the user's interests and preferences.

Preferably, users are divided into the categories of staff, patient and visitor, with the following subcategories:

- Staff (physician (department, function, specialty, interest, education, level of security, 15 favorite browser, etc.), nurse (department, function specialty, interest, education, authority for information access, favorite browser, etc.), administration (department, function, interest, etc.)).
- Patient (inpatient, outpatient, department/section (internal medicine, surgery, emergency room, cardiology, etc.), favorite browser).
- 20 - Visitor (for which patient, first time, regular, etc.).

USER PRIVILEGES CLASS:

Data of the user privileges class indicates CPR data access allowed to the user.

Preferably, this data are organized into the following categories:

- Write access or read access, and to which data.
- 25 - Full view to all CPR data of all patients.
- Full view to all CPR data of some patients.
- View only to CPR data related to user category and interest of some or all patients.
- Minimal view to CPR data of some or all patients.
- No view to any patient CPR data.

30 COMPUTER AND NETWORK CHARACTERISTICS CLASS:

Data of the computer and network characteristics class indicates the hardware and software characteristics of the user's computer and of the network connection of the computer to the system. Preferably, this data are organized into the following categories and subcategories:

- Type (SUN (Ultra 1, Sparc 20, etc.), PC (Pentium, etc.), Mac, etc.).
 - Operating system (Unix (flavor of Unix), Windows (83, 95, NT), MacOS).
 - Support for audio and video, if any.
 - Lowest bandwidth link to server (bottleneck link): ATM, fast Ethernet, T1, Ethernet,
- 5 ISDN, phone line, wireless, and so forth

DISPLAY CHARACTERISTICS:

Data of the display characteristics class indicates the characteristics of the display of the user's computer. Preferably, this data are organized into the following categories and subcategories:

- 10
- Spatial resolution (2Kx2K, 1Kx1K, XGA, SVGA, VGA, etc.).
 - Physical size.
 - Monochrome or color.
 - Modulation transfer function.
 - Amplitude resolution (number of levels of gray scale and/or of color).

15

BROWSER CAPABILITIES CLASS:

Data of the browser characteristics class indicates the software capabilities of the browser resident in the end-user device. Preferably, this data are organized into the following categories and subcategories:

- Whether or not Java[™] is enabled.
- 20
- Whether or not ActiveX is supported.
 - Which versions of HTML and HTTP are supported.
 - Which plug-ins are supported.

ROOM CHARACTERISTICS CLASS:

Data of the room characteristics class indicates any limitations due to user's

25 current room location. Preferably, this data are organized into the following categories and subcategories:

- Function of room (patient room, laboratory, film reading, library, public room, private office, home, etc.)
 - Lighting conditions (dark room, bright room, etc.)
- 30
- Audio characteristics (loud room, quiet room, etc.)
 - Audio display allowed.

Information of these classes and categories is stored in personalization database 69. This database advantageously includes a plurality of storage structures appropriate for the types of data being stored. Persistent data, which is useful across several

user sessions, can be stored either as a structured database or alternately as one or more files, or certain portions can optionally be stored in a user smart-card. Transient data, which is useful only for the current user session, can be stored as convenient, perhaps as in memory data records associated with the user session representation. Accordingly, the storage of personalization database information is preferably adapted to the type of information and to its persistence.

Information is entered in the personalization database by various methods, including, *inter alia*, through forms and from dynamically available session information. With respect to the use of forms, the system can query a user upon first accesses to enter persistent self-descriptive information. This information can be updated on user request. Preferably, the system will display a HTML form, perhaps with Java[™] applets, seeking information about the user's department or section, function, specialty, interest, education level, default browser, and so forth. Further, by using further HTML forms, the user can enter details concerning persistent preferences for screen layout, help presentation, printing formats, presentation of alerts and reminders, and so forth. In a similar manner, system administrators can enter sensitive persistent data controlling the user's authorizations and access authorities so that they are consistent with the medical institutions policies and practices.

Environment profile information is obtained from the IP address of the client, client-server browser communication, smart cards, active badges, and so forth. IP addresses of clients requesting information from the server, where they are statically rather than dynamically assigned, uniquely identify the end-user device. In such situations, the IP address of the end-user device is automatically detected by the web server when a user logs on, and then computer, network, and environment information is retrieved from a database or files at the second-tier server which includes such information indexed by IP address. Thereby, the IP address can be used to identify the computer type (e.g. SUN, PC, MAC, etc.), its add-on capabilities (e.g. sound card, video decoding hardware, etc.), its lowest bandwidth connection link to the web server (e.g. ATM, Ethernet, ISDN, wireless, etc.), the resolution of the display to which it is attached (e.g. 2Kx2K, 1Kx1K, XGA, SVGA, VGA, etc.), the location of the computer (provided it is not mobile), and constraints imposed by the location. System administrators of the hospital's computing facilities would update such system resource databases as necessary.

Alternately, the browser program running in the end-user device can detect device and browser capabilities and can communicate these to the second-tier server. Thus

the previously listed computer, add-on, and display capabilities, as well as browser capabilities (support for Java™, ActiveX, versions of HTML and HTTP, plug-ins, etc.), can be made available. This is possible even where the IP address is not a reliable indicator of such information. Also, information in a smart card carried by a user can be read to provide certain user attributes as well as to prove user identity.

Rules Modules - Personalization Family

The rules of this invention, which are separately stored in rule databases and are retrieved by the business-server objects and the rules engine, are advantageously grouped into and retrieved as rule modules, which are collections of rules and rule-sets directed to similar situational analysis. Further, for purposes of description, the rules modules are assigned to two families of rules modules, namely decision support rules module family 71 (Fig. 3) which are directed to providing decision support services to end-users, and personalization rules module family 72 which are directed to personalizing and formatting system data and results.

The personalization rules module family is described first followed by the decision support rules module family. The following are examples of constraints imposed in personalizing the content:

- The specialty, such as cardiology, of a physician user should be taken into account so that the physician receives only the news and information which is likely to be of interest.
- Information should only be provided to users with sufficient access privileges.
- Rooms that are meant to be quiet rooms such as reporting rooms or library locations should not be exposed to end-user devices playing audio files.
- A computer that does not have a sound card should not receive audio files in any case.
- A laptop with a low-speed modem and a small screen should not be exposed to large movie files or large images.
- A browser that does not support ActiveX should not be exposed to ActiveX components.

Personalized content is not only more interesting and relevant to the user. It also makes more efficient usage of network bandwidth and system resources, reduces server load, and can also reduce document retrieval latency.

Accordingly, personalization rules module family 72 preferably includes at least an alerts & reminder rule module, an access privileges rules module, a platform rules module, a graphic user interface ("GUI") layout rules module, a help rules module, a help rule module, an order & forms rules modules, a printing rule module, and such other personalization rules modules as a medical institution finds advantageous.

ACCESS PRIVILEGES RULES MODULE:

The rules of the access privileges rules module determine which user is authorized to view which information concerning which patients. First, access privileges depend on federal and state regarding patient confidentiality law. Each medical institution hospital may also have local protocols for ensuring patient confidentiality, data integrity, and security (digital signatures, authentication, and document alteration prevention techniques). The system administrator of the hospital is responsible for maintaining such policies and rules for information access.

Access privileges also depend on the user function, role and relationship to the patient (i.e. patient's attending physician, consulting physician, attending nurse, etc.). For example, not only do different occupations/specialties need different "views" of the CPR which are tailored to their needs but different patient relationships may influence the level of detail presented in sensitive areas. For example, all physicians who treat a patient may see that the patient is undergoing psychiatric treatment, but the details of any psychiatric treatment may be viewable only by the attending psychiatrist and the patient. Also, access to records for certain "VIP" patients (politicians, actors, etc.) may be further restricted than for normal patients, due to the increased potential for adverse publicity and blackmail. Patients should be able to see their own CPRs, in full detail. The same is also true for legal guardians of underage or legally incapable patients.

All users should have a default log-in which has minimal privileges, but is based on location. Therefore, any health care provider inside a hospital may be able to see a summary CPR for any in-patient without a special log-in (other than identifying themselves as health care providers, for example via smart card ID). However, this capability would not be available from outside the firewall guarding the integrity of the intranet.

PLATFORM RULES MODULE:

The rules of the platform rules module determines how content should be personalized formatted for a particular end-user device connected on a particular network link. Such personalization and formatting is independent of user requested personalization because many users (physicians, in particular) may have many types of equipment and network links, e.g. at the office, the hospital, and at home, which have widely varying abilities. Further, the system can include end-user devices which are not assigned to particular end-users for public use, e.g., by referring physicians visiting their in-patients.

Therefore, the business-server objects, acting on recommendations of the platform rules module, personalizes and formats information presented to be consistent with

an end-user device's location and capabilities. As discussed, some information on these capabilities is gathered at user login. The content can be dynamically generated and automatically customized to match the capability of the client browser, display, and link bandwidth.

5 For example, images and sound files are personalized and formatted (full resolution or minified) and compressed for transmission (none, loss less, lossy/quality) according to the link bandwidth and the capabilities of end-user devices, so that a user need not wait for large transfers at locations with low speed connections. If the display of an end-user device can not handle high-resolution images, then low-resolution images are presented
10 on this device, even if the user has requested otherwise. Likewise, if the network connection of the user is slow, then low resolution images should be transmitted, unless the user has indicated acceptance of the necessary wait for full resolution images.

 Concerning specifically images and video, end-user devices with low-bandwidth connections and/or low-resolution displays need low-resolution images and
15 lower frame-rate video. When considering heterogeneous networks the lowest network bandwidth link between the server and the clients is the limiting factor. Scaling and layered video coding schemes are one method which can be used to multicast one single compressed video stream across heterogeneous networks, computers, and displays. The routers can then send the appropriate number of compressed video layers to the appropriate machines for
20 software or hardware decoding. For example, a high-end workstation with a high resolution display and high bandwidth connection will receive the base layer plus all the enhancement layers. An intermediate computer with a moderate resolution display and a moderate bandwidth connection can receive the base layer plus some of the enhancement layers. A low-end workstation with a low resolution display and a wireless connection, however, will
25 receive only the base layer.

GUI LAYOUT RULES MODULE:

 The rules of the GUI (graphical user interface) rules module determine how content to be presented should be arranged on the display screen of an end-user device for a particular end-user. These rules can interpret user preferences stored in the personalization
30 database. Alternatively, a user may personalize the rules themselves.

 Rules of this module can determine the overall layout of an end-user device screen display. For example, screen displays can be partitioned to accommodate different information categories in different regions. A large portion of the screen will be reserved for personalized and formatted information requested by the user. Another portion of the screen

will be reserved for general information which should be seen by everyone at a medical institution. Finally, a third portion will be reserved for personal information, such as, for example, the web sites and content which the user visits most frequently.

Further, rules of this module can determine how these screen portions are formatted in detail. For example, a user can specify a flag to view numerical results in tabular form or in a graphical plot. The user can request not to view images. The user can specify the positioning of information items (for example, show the present lab results next to the previous lab result, with the most recent lab results on the left).

HELP RULES MODULE:

The rules of the help rules module determines how the on-line help system of this invention is personalized to provide the most efficient assistance to an end-user.

Preferably, the help and explanation provided can be personalized based on the level and experience of the user. The user information that is stored in the personalization database can have a "user level" attribute for each user to differentiate expert CPR users from novice ones. Alternatively, the system can use the audit trails and/or track users to figure out the level of each user.

Also preferable, is that the rules of the help rules module adapt the screen display to create a personalized help system. Which help system components are displayed can be context sensitive and depend on the task that the user is performing (e.g. viewing, reporting, ordering, updating, etc.) and the information category that being acted on (Lab, pharmacy, radiology, etc.).

ORDERS & FORMS RULES MODULE:

The rules of the orders & forms rules module determines which users have privileges to access the system capabilities to fill in orders and submit admitting and other forms from end-user devices. The present invention provides the capabilities for the large number of forms and orders relating to patient care in a medical institution to be filled-in from end-user devices. Different users have different roles and capabilities for filling-in forms. For example, only physicians can enter order for medications or treatments. The rules of this module control access to forms and the ability to fill-in or modify them based on user attributes similar to those governing general access privileges.

PRINTING RULES MODULE:

The rules of the printing rules module determines print formats in accordance with a user's printing preferences stored in the personalization database. Therefore, upon a user print request, the business-server objects can print pages personalized and formatted

according to the user's specifications. For example, these rules can specify that images not be printed or be printed only at a reduced size. The users can change their own printing preferences.

ALERTS & REMINDERS (NOTIFICATION) RULES MODULE:

5 The rules of the alerts & reminders rules module (also called the notification rules module) of the personalization family determines personalization, formatting, and presentation of the results generated by the alerts & reminders rules module of the decision support family. Personalization and formatting rules determine the physical layout and screen presentation of alerts and reminders. Presentation rules can also determine the manner
10 of transmission of these messages, which may be a function of the priority indicated for the message. For example, a rule of this module may specify: IF the message is an urgent alert, THEN page me and send me e-mail with the alert appropriately formatted. Another exemplary rule may specify: IF the message is a reminder, THEN send me formatted e-mail only.

15 Rules Module - Decision Support Family

Decision support functions can be easily integrated into the structure of business-server objects 67 (Fig. 3), rules engine 68, and rules modules by simply introducing additional rules modules which provide decision support reasoning capabilities. Such decision support services are advantageous in the context of a medical institution. Therefore,
20 decision support rules module family 71 typically includes at least an alerts & reminder rule module, a diagnosis and interpretation rules module, and a drug interaction rules module.

The knowledge-base of decision support rule modules is typically obtained and maintained by domain experts. One of skill in the art will understand how other decision support services that a medical institution finds advantageous can be implemented in
25 additional decision support rules modules to be added to this family and knowledge base.

ALERTS AND REMINDERS RULE MODULE:

30 The rules alerts & reminders rules module of the decision support family determines decision support functions available to notify care givers of generally urgent or notable events. In more detail, decision support alerts are generated to draw the user's attention to urgent events, e.g. abnormal test results, urgent orders or medications not processed, possible drug interactions, etc. For example, such a rule may be: IF white cell count is greater than 20,000, THEN generate an alert message. Decision support reminder messages are generated for more routine and less urgent events, or can be entered by the user for latter display as a "tickler" message.

These rules are patient, episode and user specific. They allow only certain classes of patients to be examined with a certain frequency for certain types of events to generate certain alerts and reminders. For example, a physician may wish to routinely scan only diabetic patients for abnormal glucose trend events and only patients being administered cancer chemotherapeutic for dangerous white cell count events. Rules of the "Alerts and Reminders" module allow a user to choose which events rank as more urgent alerts and which events are adequately addressed by less urgent reminders. Further, these rules can also specify which alerts require urgent attention, and which event are not urgent and are adequately addressed by reminders. For example, some physicians may want to be alerted about certain events on a particular patient whereas others may only want reminders. Alerts and reminders can optionally be assigned a priority. For example, an alert may be marked as "urgent".

DIAGNOSIS & INTERPRETATION RULES MODULE:

The diagnosis & interpretation rules module includes rules representing diagnostic support and interpretation of particular conditions and results. These rules can scan a CPR of a particular patient and provide suggested diagnoses or interpretations.

Such rules are well known in the art. For example, the following can be coded in a convenient rules language: IF the patient has fever, sore throat, and on physical examination has white exudate in the posterior region of the pharynx and enlarged lymph nodes in the anterior cervical region, and laboratory tests revealing a positive streptococcal antigen antibody blood level, THEN the patient has streptococcal pharyngitis.

DRUG INTERACTION RULES MODULE:

The drug interaction rules module includes rules representing known adverse drug interactions and can provide warnings of potential adverse interactions in a particular patient given information on the medications of that patient.

Such rules are well known in the art. For example, the following can be coded in a convenient rules language: IF a patient is taking a monoamine oxidase inhibitor, such as deprenyl, and antidepressants, such as Prozac, THEN the patient can have a fatal reaction.

Prioritization and Update of Rule Modules

As discussed previously, in order to achieve predictable and efficient rule processing, rule and rule modules are prioritized and are processed by the rules engine in priority order. The highest priority rule modules are the access privileges and the orders & forms rule modules. For example, the GUI rules module has a lower priority, because, if a user is not authorized to view certain types of information for a particular patient, then that

information should not be accessible even if the user indicates otherwise in the screen layout preferences in the GUI rules module. Similarly, if a user is not authorized to order for a particular patient, then that user should not be able to display orders and forms menus on a display screen.

5 At the next level of priority modules is the platform rules module. If the capability of the platform can not handle certain information, *e.g.*, high-resolution images, then rules which deal with the formatting and personalization of that information, *e.g.*, preferences for how to display high resolution images, need not be evaluated.

 Therefore, at the lowest priority level are the GUI layout, the alerts &
10 reminders, the help, and the printing rules modules.

Push Technology for Rules Updating

 Where the rules modules of this invention are duplicated on several second-tier server systems or where certain of the rules modules have been downloaded to capable end-user computers, it is preferable to update in a coordinated fashion all copies of the rules
15 modules. It is similarly preferably to update other software components that are necessarily duplicated on various computers and device of this system. Such updates are done by installation means.

 To perform such coordinated update, push technologies are advantageously employed as installation means. Push technologies are generally taken herein to mean
20 technologies that maintain information on where software components are or should be in a system, and, when presented with a new software component or an updated existing software component, will automatically place this component in the appropriate locations in a system. For example, push technologies can copy an updated rules module to all middle-tier server systems and to all end-user devices to which it has been downloaded.

25 One preferable push technology is available in the Castanet System of Marimba, Inc. (<http://www.marimba.com>).

 Push technologies have further applications for providing medical news channels and information, medical education, medical reference information, and so forth to end-users at end-user devices from news servers on the Internet. Where such broadcast is
30 provided, the type and formatting of the broadcast information can be controlled by additional personalization rules modules in the manners previously described.

 The systems and methods of his invention are not limited to the preferred domain of the activities of medical institutions. One of skill will understand how the same systems and methods can be applied to the activities of other institutions, such as, for

example, banks, insurance companies, airlines, hotels, manufacturing corporations, and so forth.

In order to create a business rule application according to this invention in another domain, the first steps are to create rules and rule-based business server objects.

5 Different rule-based business-server objects can be implemented for different rule modules to improve scalability and maintenance. In a medical application, different business-server objects can be created for handling help, for orders & forms, for diagnosis & interpretation, and so forth. The rules can also be compiled to create Java[™] classes.

10 It should now be appreciated that the objects of the present invention are satisfied. While the present invention has been described in particular detail, it should also be appreciated that numerous modifications are possible within the intended spirit and scope of the invention.

All references cited herein are incorporated herein by reference in their entirety and for all purposes to the same extent as if each individual publication or patent or
15 patent application was specifically and individually indicated to be incorporated by reference in its entirety for all purposes.

CLAIMS:

1. An object-oriented system for computerized patient record (CPR) presentation to a user at an end-user device, the object-oriented system for implementation on computers connected by a network, the system comprising:

one or more medical-records-server objects comprising CPR-request methods
5 that input requests for CPR data, access requested data in CPR databases, and return requested CPR data,

a presentation application resident in the end-user device that accepts user requests, invokes methods of business-server objects with parameters representing user requests, and displays responses returned by the business-server object methods,

10 one or more business-server objects comprising the business-server object methods invoked by said presentation application,

wherein the business-server object methods process input parameters to determine if CPR data is to be obtained, and if so which CPR data, authorize access to the determined CPR data, invoke the CPR-request methods of said medical-records-server
15 objects to obtain authorized CPR data, format a response including any returned CPR data, and return the response to said presentation application, and

wherein the authorizing and formatting are responsive to a plurality of rules retrieved from a rule database and to personalization data retrieved from a personalization database, the rules comprising (i) access-control rules for authorizing access to CPR
20 information and (ii) presentation-control rules for guiding formatting of responses.

2. The system of claim 1 further comprising one or more rules engines that processes rules and personalization data to determine rules recommendations, and wherein the authorizing and formatting of the business-server object methods are responsive to rules
25 recommendations returned from calls to said rules engines.

3. The system of claim 2 wherein one or more rules are downloaded to the end-user device, and wherein display by the presentation application is responsive to

recommendations returned by calls to a rules engine resident in the end-user device upon processing the downloaded rules.

4. The system of claim 1 wherein related rules are grouped for storage and retrieval in the rules database into rule modules, which are separately stored and retrieved in the rules database.

5. The system of claim 4 further comprising installation means for installing new or updated rules modules in the rules database.

6. The system of claim 4 wherein the rules modules have a priority, and wherein retrieval of the rules modules from the rule database by the business-server objects or by the rules engines is responsive to the priority of the rules modules.

7. The system of claim 4 wherein the access-control rules further comprise an access-privileges rules module that authorizes access to CPR data concerning a patient in dependence on personalization data including characteristics of the role of the user in an institution, of the role of the user with respect to the patient, and of the access privileges of the user.

8. The system of claim 4 wherein the access-control rules further comprise an orders-and-forms rules module that authorizes a user to submit orders and forms relating to a patient.

9. The system of claim 4 and wherein the presentation-control rules further comprise a platform rules module that recommends response formatting in dependence on data including the characteristics of the end-user application, the end-user device, and the network.

10. The system of claim 4 wherein the presentation-control rules further comprise a GUI-layout rules module that recommends response formatting in dependence on personalization data including graphical-user-interface layout preferences of a user.

11. The system of claim 4 wherein the presentation-control rules further comprise a help rules module that recommends formatting and display of help information in dependence on personalization data including characteristics of the experience of a user using the system.

5

12. The system of claim 4 wherein the presentation-control rules further comprise a printing rules module that recommends formatting of printed responses in dependence on personalization data including characteristics of the printing preferences of a user.

10 13. The system of claim 4 wherein the rules further comprise decision-support rules for providing decision-support recommendation to a user.

14. The system of claim 13 wherein the decision-support rules modules further comprise a first alerts-and-reminders rules module for providing alerts and reminders
15 messages concerning conditions of one or more patients, and wherein the presentation-control rules further comprise a second alerts-and-reminders rules module for recommending the formatting and presentation of the alerts and reminders messages

15. The system of claim 13 wherein the decision-support rules further comprise a
20 diagnosis-and-interpretation rules module comprising rules for providing recommended diagnoses and interpretations of a patient's condition.

16. The system of claim 13 wherein the decision-support rules further comprise a
25 drug-interaction rules module that provides recommendations on possible drug-drug interactions.

17. The system of claim 1 wherein said presentation application comprises a Java[™]-enabled browser program which requests and displays data formatted according to the hypertext markup language (HTML), and wherein the object-oriented system further
30 comprises a hypertext transfer protocol (HTTP) server for providing requested HTML-formatted data and Java[™] applets to said presentation application.

18. The system of claim 1 wherein the end-user device comprises a computer with a graphical display device and processing means, and wherein the processing means comprises an ORB and said presentation application.

5 19. The system of claim 1 wherein the end-user device comprises a personal digital assistant or a television set.

20. The system of claim 1 further comprising object request brokers (ORBS) which reside on each computer of the system and which provide for remote method
10 invocations across the network of object methods.

21. A method of presenting computerized patient records (CPR) on an end-user device by an object-oriented system implemented on computers connected by a network, the method comprising:

15 accepting a user request and invoking one or more methods of business-server objects, wherein parameters input to the business-server object methods represent the user request, and wherein said accepting and invoking are performed by a presentation application in the end-user device,

determining if CPR data is to be obtained for the user, and if so which CPR
20 data, wherein said determining is performed by the business-server objects according to the input parameters,

authorizing the determined CPR data according to authorization
recommendations returned from processing both of access-control rules retrieved from a rules database and also of personalization data retrieved from a personalization database,
25 wherein said authorizing is performed by the business-server objects,

invoking one or more methods of medical-records-server objects, wherein parameters input to the medical-records-server objects represent the authorized CPR data, wherein the medical-records-server objects return the authorized CPR data retrieved from CPR databases, and wherein said invoking is performed by the business-server objects,

30 formatting and returning a response to the presentation application, wherein the response includes returned CPR data, wherein the formatting is according to formatting recommendations returned from processing of presentation-control rules retrieved from the rules database and personalization data retrieved from the personalization database, and wherein the formatting and returning is performed by the business-server objects, and

displaying the returned response to the user by the presentation application.

22. The method of claim 21 wherein rules processing further comprises calling a rules engine by the business-server object methods, and wherein the rules engine returns to the business-server object recommendations resulting from rules processing.

23. The method of claim 21 further comprising, prior to said displaying, further formatting of the returned response by the presentation application.

24. The method of claim 21 wherein related rules are grouped for storage and retrieval in the rules database into prioritized rules modules, and wherein retrieval of rule modules is responsive to rule module priority.

25. The method of claim 24 further comprising a step of processing decision-support rules retrieved from the rules database, wherein the decision-support rules processing is responsive to the parameters input to the business-server object methods, and wherein the decision-support recommendations returned from the decision-support rules processing are included in the formatted response.

26. The method of claim 25 wherein the decision-support rules comprise one or more rules modules selected from an alerts-and-reminders rules module, or a diagnosis-and-interpretation rules module, or a drug-interaction rules module.

27. The method of claim 24 wherein the access-control rules comprise one or more rules modules selected from an access-privileges rules module or an orders-and-forms rules module.

28. The method of claim 24 wherein the presentation-control rules comprise one or more rules modules selected from a platform rules module, or a GUI layout rules module, or a help rules module, or a printing rules module.

29. The method of claim 21 wherein at least one presentation application, at least one business-server object, and at least one medical-records-server object reside on different computers, and wherein said steps of invoking methods further comprise passing of method

invocations between different computers by object request brokers resident on each computer.

30. An object-oriented computer program for performing the method of claim 21.

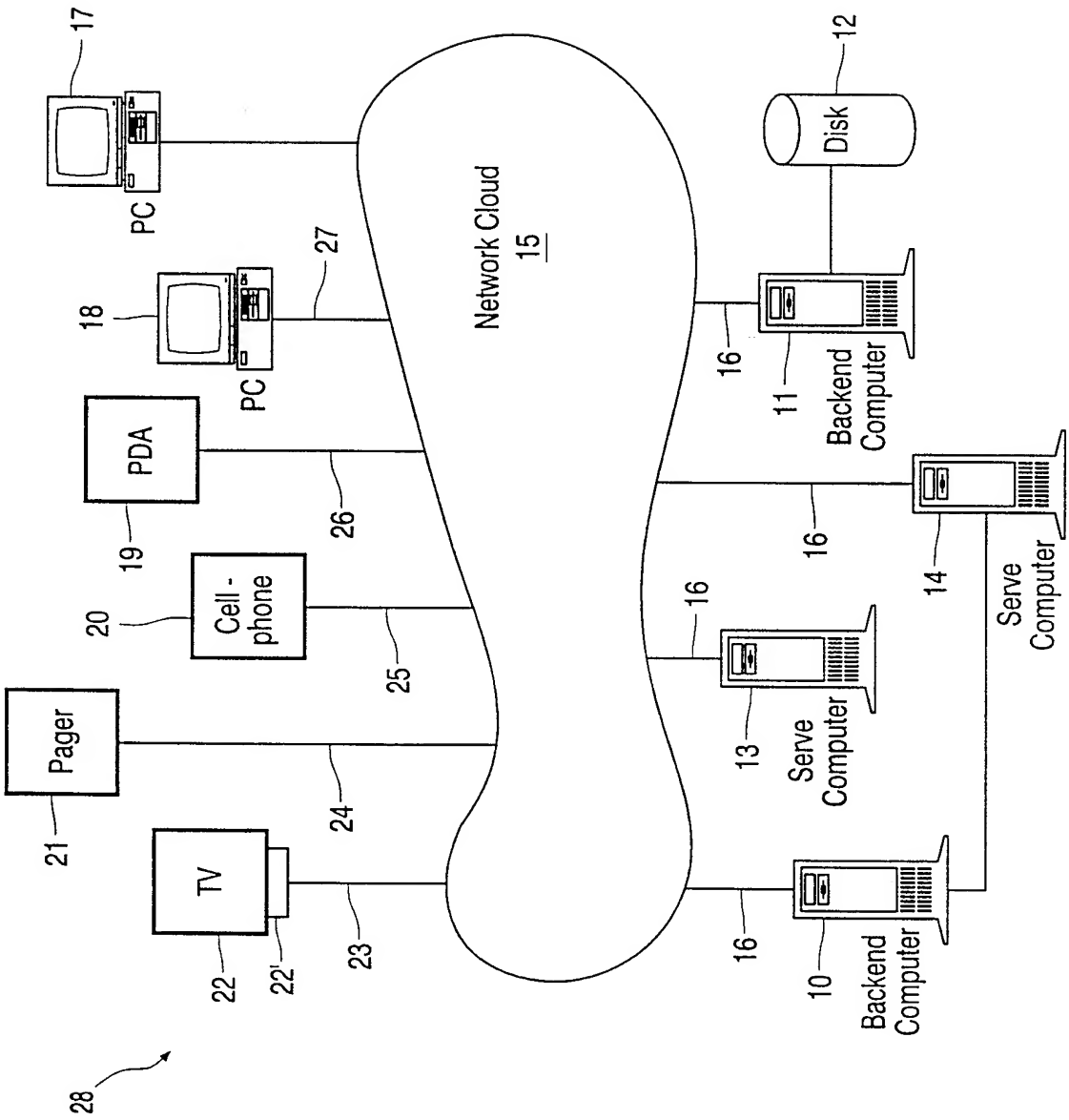


FIG. 1A

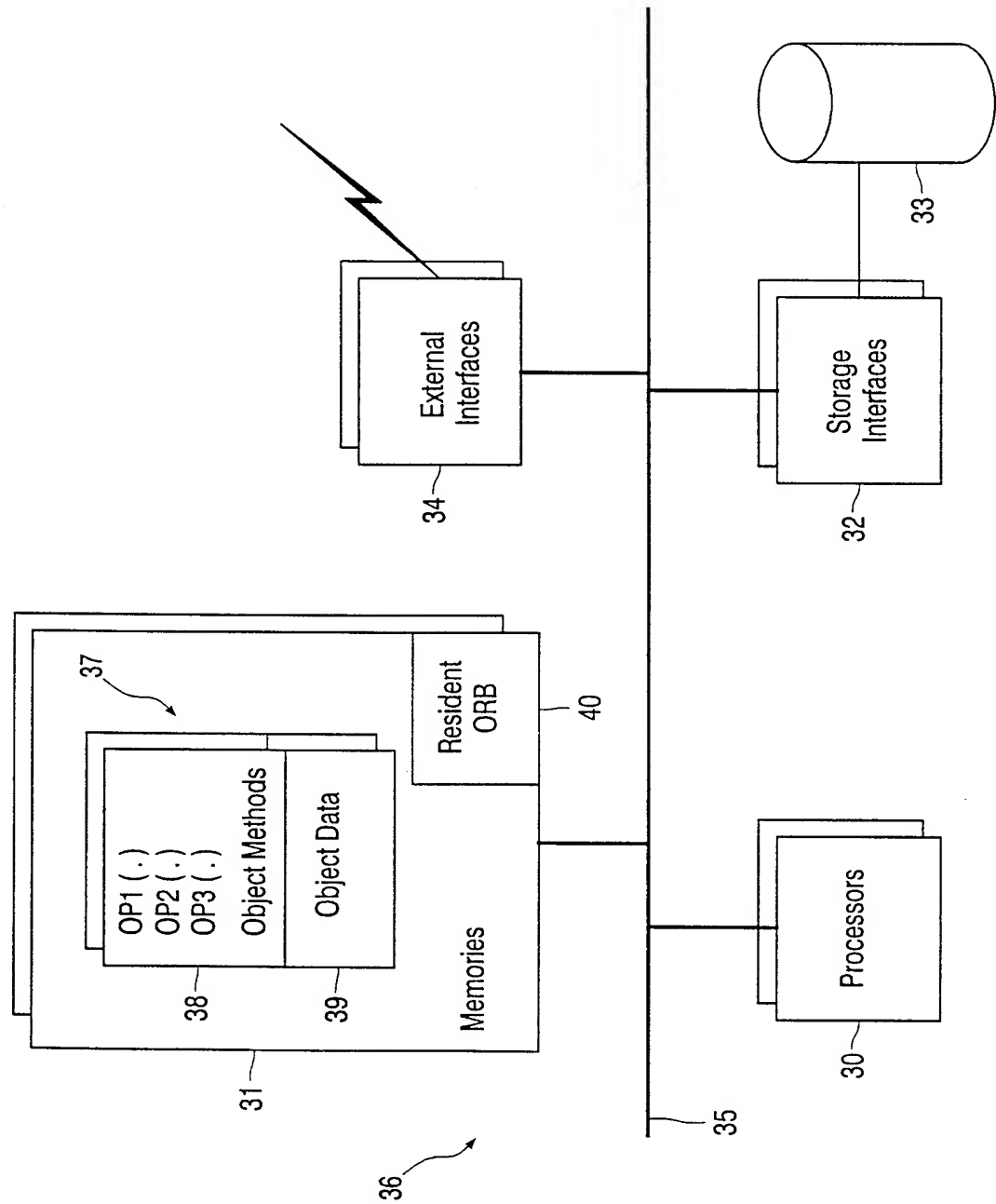


FIG. 1B

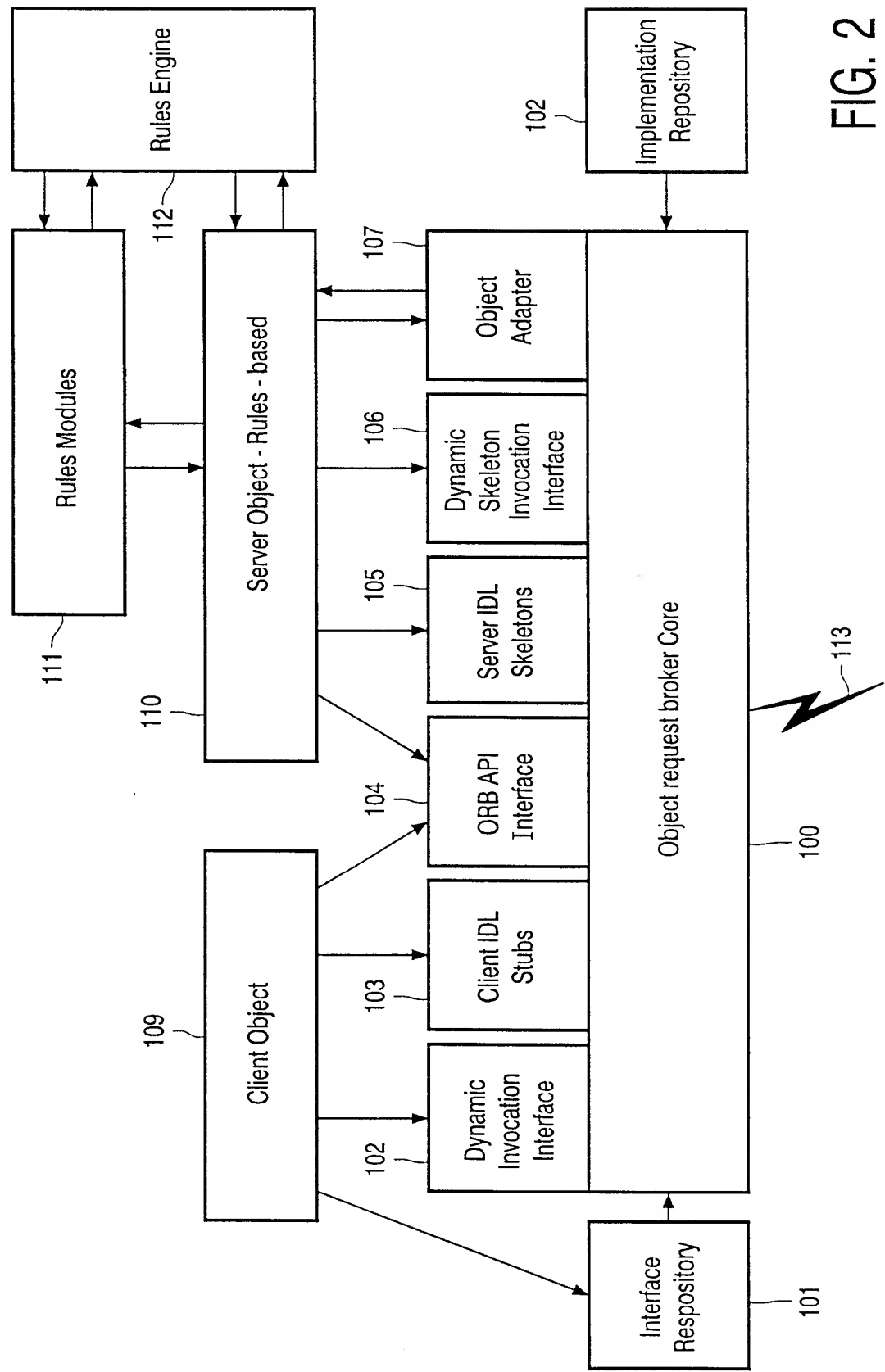


FIG. 2

4/5

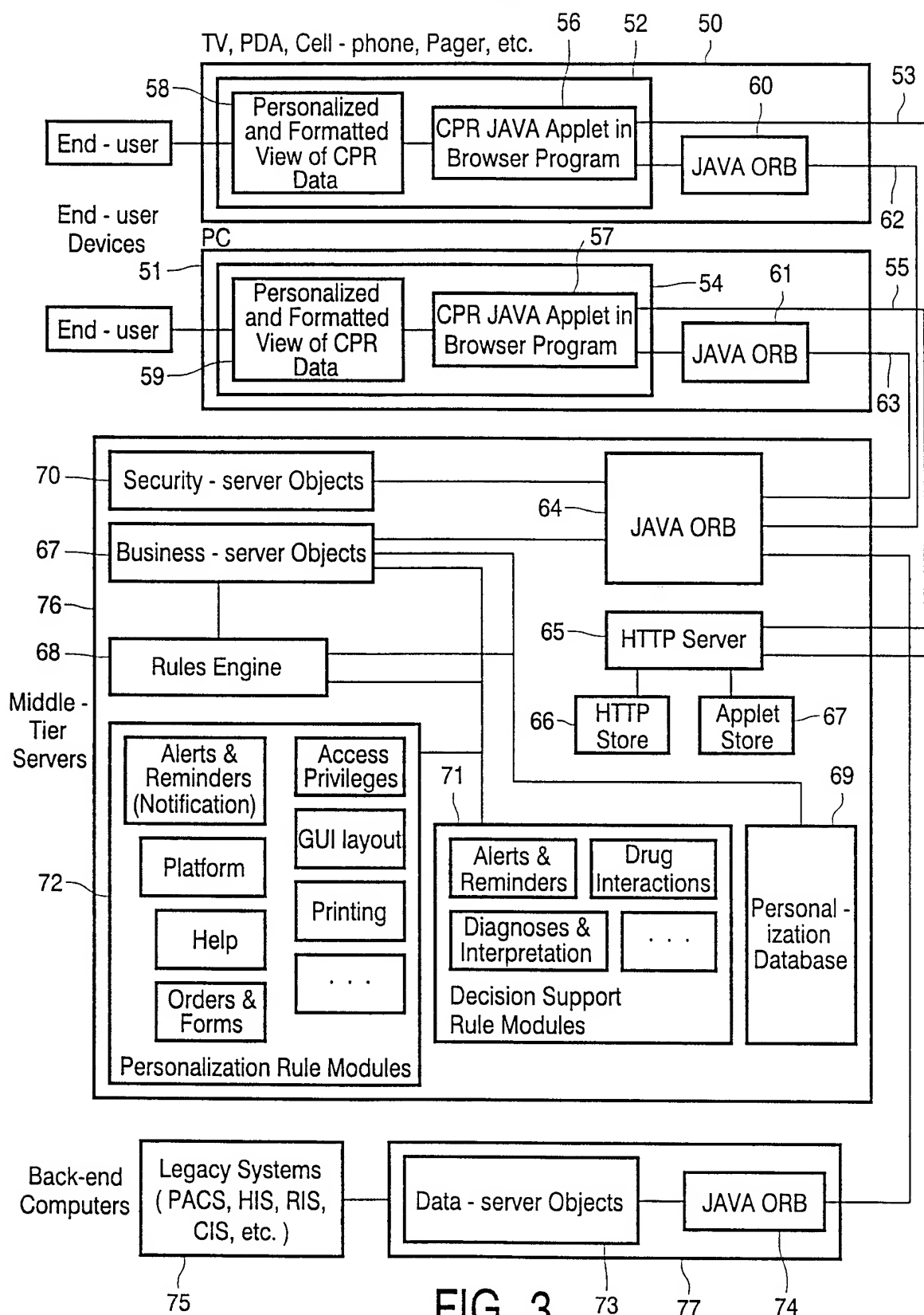


FIG. 3

5/5

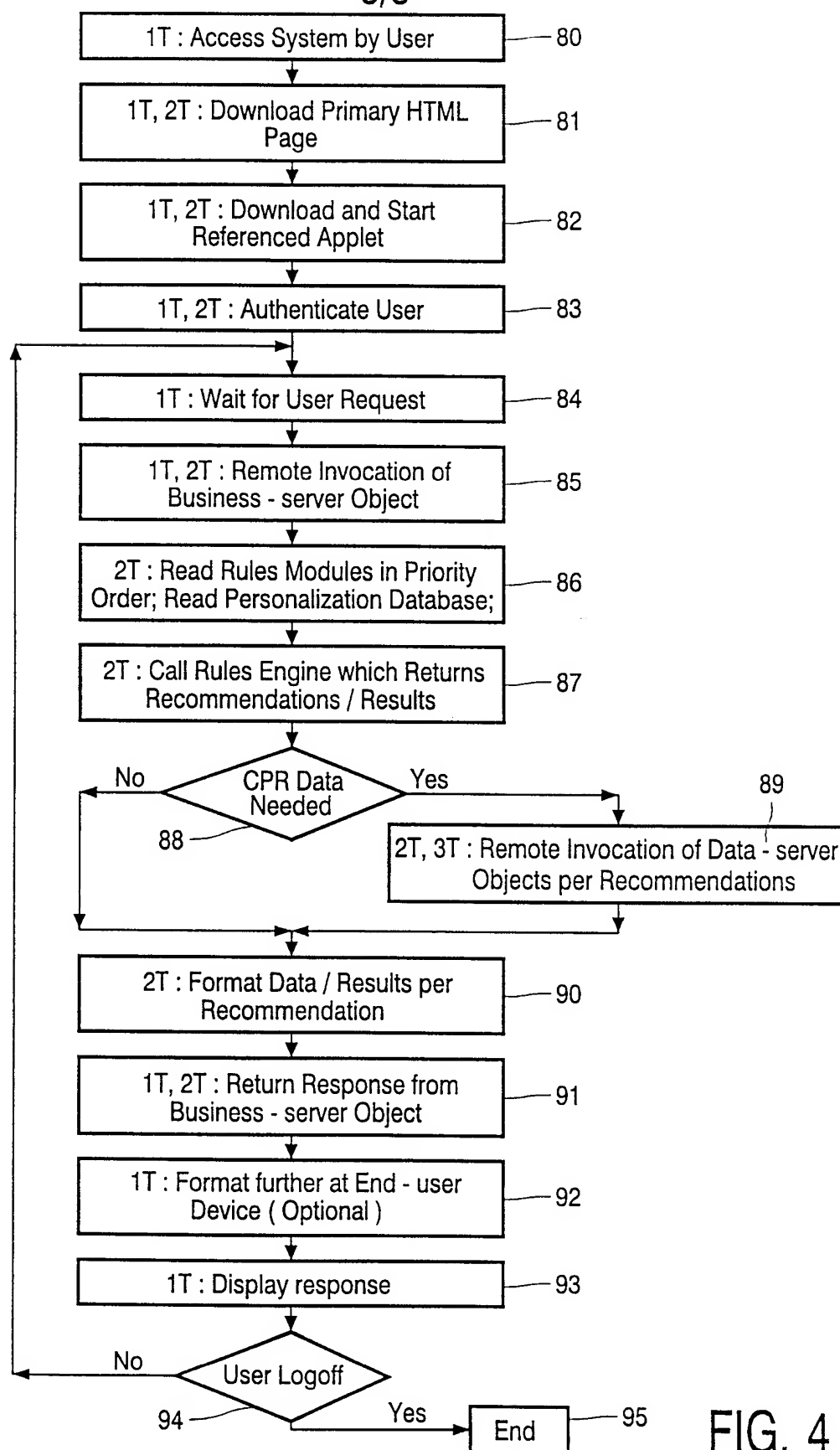


FIG. 4